REMARKS

I. Status of Claims for Amendment under 37 CFR §1.111

Claims 1, 2, 4-12, 14-20, 22-24, 26-35, 37 and 38 are pending in this application for this Amendment filed in response to the October 3, 2007 Non-Final Office Action in which claims 1-24 and 26-38 were pending. No claims are allowed, and the previous allowability of claims 22, 25, 27 and 30-32 in the May 3, 2007 Office Action appears to have been withdrawn without explanation. Reconsideration based on the following remarks is respectfully requested.

II. Amendments under 37 CFR §1.121

By this Amendment, claim 3, 13, 21 and 36 are cancelled without prejudice to or disclaimer of the subject matter contained therein, and claims 1, 4-7, 14, 17 and 34 are amended in accordance with 37 CFR §1.121(c)(2).

Claim 1 is amended to recite features from claims 3 and 13, claim 17 is amended to recite features from claim 21, claim 34 is amended to recite claims from claim 36, and claims 4-7 and 14 are amended to correct minor informalities. No new matter is added by any of these amendments.

III. Anticipatory Rejection under 35 U.S.C. §102

The Office Action rejects claims 1-24 and 26-38 as being allegedly anticipated under 35 U.S.C. §102(e) over U.S. Patent 5,974,254 to Hsu *et al.* (hereinafter "Hsu"). This rejection is rendered moot with respect to claims 3, 13, 21 and 36, and is respectfully traversed for the remaining claims.

A. Description of Inventive Features: Applicants' claimed features are directed to a multiple-user graphical programming and analysis environment program (108) that includes (as provided in claim 17) graphically represented code objects (208, 212), graphically represented inter-code object connections (214) and at least one application program (210) within a white board area (202). Each code object (208) is created by a user (206) and accessible by other users in accordance with security privileges of the other users. Each inter-code object connection (214) represents data transfer between a pair of code objects (212). The application program is

composed of at least one chain of the code objects interconnected by the inter-code object connections. The graphical white board area (202) provides a region within which the code objects are definable and movable and the inter-code object connections are creatable. The application program is executable within the graphical white board area. These features are described in the specification, for example, at page 7 line 13 – page 9 line 2 (corresponding to paragraphs [0023] – [0025] in U.S. Patent Application Publication 2005/0138601) and FIG. 2.

Each inter-code object connection (214) terminates on one of an edge and an interior of one of the code objects (212). These features are described in the specification, for example, at page 10 lines 10-20 (corresponding to published paragraph [0029]. Each code object (300) includes a data interface indicating first data (302) to be input into the code object and second data (306) to be output by the code object, and internal logic (304) to generate the second data from the first data. These features are described in the specification, for example, at page 11 line 8 – page 12 line 13 (corresponding to published paragraphs [0031] – [0033]) and FIG. 3.

Applicants' claimed features are also directed to methods (as provided in claims 1 and 34) for operating in a multiple-user graphical programming and analysis environment. One method provides for steps described in the specification, for example, at page 16 line 10 – page 17 line 23 (corresponding to published paragraphs [0046] – [0049]) and FIG. 6. The steps include: accessing by a user (206) a graphical programming and analysis environment program (108) that other users are already currently accessing (602), generating by the user graphically represented code objects (300) within the environment program (604), graphically chaining together code objects by the user within the environment program (610) and assembling application programs by the user within the environment program, each application program composed of the code objects as have been chained together (612).

The code objects generating step further includes steps that for each code object, the user determining a data interface indicating first data (302) to be input into the code object and second data (306) to be output by the code object (606), and the user determining internal logic (304) to generate the second data from the first data (608). The chaining step includes chaining together the code objects generated by the user and code objects generated by the other users to which the user has access based on security privileges accorded to the user, to yield inter-code object

communication by inter-code object connections. Each inter-code object connection terminates on one of an edge and an interior of one of the code objects. Each application program is composed of the code objects that have been chained together.

The other method provides for steps described in the specification, for example, at page 18 line 1 – page 19 line 12 (corresponding to published paragraphs [0050] – [0053]) and FIG. 7. The steps include: providing asynchronous access to multiple users to a graphical programming and analysis environment program (108) visually represented as a white board (702), allowing each user of the multiple users to generate graphically represented code objects within the environment program (704), allowing each user access to the code objects of other users of the multiple users based on security privileges accorded to the user (714), allowing each user to have the code objects of the user be chained to the code objects of the other users to which the user has access to yield inter-code object communication by inter-code object connections (716), and allowing each user to execute application programs composed of the code objects as chained together within the environment program (718). The code objects generating step further includes allowing said each user to instantiate one or more code objects (706), allowing said each user to determine an internal logic for each code object (708), allowing said each user to determine first data to be received by said each code object (710), and allowing said each user to determine second data to be sent by said each code object (712). Each inter-code object connection terminates on one of an edge and an interior of one of the code objects.

B. Applied Reference: Hsu does not teach or suggest the methods recited in claims 1 and 34, and further fails to teach or suggest the multiple-user graphical programming and analysis environment program recited in claim 17. Instead, Hsu discloses a method for detecting differences of objects between graphical programs. See col. 4 line 26 – col. 5 line 25 of Hsu. In the background, Hsu characterizes U.S. Patent 4,901,221 to Kodoski *et al.* as providing a block diagram editor with which to create text-based code using a graphically based programming environment. See col. 2 lines 34-57 of Hsu. In particular, Hsu teaches a differences detection software program called "diff" in flowchart form to compare two graphical programs.

In Hsu, diff determines differences for the block diagrams, the user interface panels and the program attributes. After receiving the programs in step 101 and graphing the diagram and

panel in step 102, the objects between the programs are matched in step 103, after which the differences between the diagrams, the panels and the attributes are respectively determined in steps 106, 108 and 110. See col. 7 lines 55-64, col. 9 line 5 – col. 10 line 8 and Fig. 3 of Hsu. Further, Hsu teaches a graph data structure using block diagrams from step 102. The structure tabulates an adjacency matrix 322 to cross-reference edges and nodes, a vertex list 324 to identify objects and an edge list 326. See col. 11 lines 2-49 and Fig. 6 of Hsu.

The interface panel includes a directed acyclic graph of the objects having sets of vertices and edges, with every edge connecting exactly two vertices. The block diagram includes graphical code to perform operations using function nodes (*e.g.*, operators, loops, communications, etc.) for objects denoting panel controls and indicators. See col. 8 line 41 – col. 9 line 4 of Hsu. None of these teachings in Hsu for graphical difference determination remotely corresponds to chaining together graphical objects with inter-code connections on a white board area or to providing internal logic within each such object for processing input and output data, as provided in Applicants' claimed features.

The Office Action at page 9 analogizes the unit under test (UUT) 23 to the white board as featured in the claims. See col. 7 lines 26-34 and Fig. 1 of Hsu. Applicants submit that no rationale exists to infer such an equivalence. The UUT couples various instruments 14, 16, 21 into an instrument control system 10. By contrast, a "white board" is used to "visually represent the environment" in the context of Applicants' specification at page 6 line 2 – page 7 line 2 (corresponding to published paragraph [0021]) and FIG. 1. In addition, an Internet-available definition from http://www.answers.com/topic/whiteboard?cat=technology provides that "The electronic equivalent of chalk and blackboard, but between remote users. Whiteboard systems allow network participants to simultaneously view one or more users drawing on an on-screen background or running an application... Only one user is actually running the application... A copy of the current application window is pasted onto the whiteboard, which then becomes a static image for interactive annotation..." Moreover, the Office Action at page 11 identifies the matching edges between object pairs at page 13 lines 12-21 in Hsu with the chat area (204) in claim 33. Applicants find no such analogy.

C. Statutory Deficiency: A claim must be literally disclosed for a proper rejection under §102. This requirement is satisfied "only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." See MPEP §2131. Applicants assert that the Office Action fails to satisfy this requirement with Hsu.

Consequently, all the claims are in condition for allowance. Thus, Applicants respectfully request that the rejection under 35 U.S.C. §102 be withdrawn.

IV. Conclusion

In view of the foregoing amendments and remarks, Applicants respectfully submit that this application is in condition for allowance. Favorable reconsideration and prompt allowance are earnestly solicited.

Should the Examiner believe that anything further is desirable in order to place this application in even better condition for allowance, the Examiner is invited to contact Applicants' undersigned representative at the telephone number listed below.

Respectfully submitted,

Gerhard W. Thielman
Registration No. 43,186

Date: January 2, 2008

DEPARTMENT OF THE NAVY
Naval Surface Warfare Center – Dahlgren Division
Office of Counsel – Code XDC1
17632 Dahlgren Road, Suite 158
Dahlgren, Virginia 22448-5110
Telephone: (540) 653-8061 **Customer No. 23501**

DEPOSIT ACCOUNT USE AUTHORIZATION
Please grant any extension necessary for entry;
Charge any fee due to our
Deposit Account No. 50-0967